Examensarbete

HÖGSKOLAN I SKÖVDE

A COMPARATIVE STUDY FOR CLASSIFICATION ALGORITHMS ON IMBLANCED DATASETS

An investigation into the performance of RF, GBDT and MLP

Examensarbete inom huvudområdet informationsteknologi
Grundnivå nivå 30 Högskolepoäng
Vårtermin År 2020

David Stenvatten

Handledare: Eva Söderström
Examinator: Mikael Berndtsson

Abstract

In the field of machine learning classification is one of the most common types to be deployed in society, with a wide amount of possible applications. However, a well-known problem in the field is classification is that of imbalanced datasets. Where many algorithms tend to favor the majority class and in some cases completely ignore the minority class. And in many cases the minority class is the most valuable one, leading to underperforming and undeployable implementations.

There are many proposed solutions for this problem, they range from different algorithms, modifications of existing algorithms and data manipulation methods. This study tries to contribute to the field by benchmarking three commonly applied algorithms *(Random forest, gradient boosted decision trees and multi-layer perceptron),* in combination with three different data-manipulation methods *(*oversampling*, undersampling and no data manipulation)*. This was done through experiments over three differently shaped datasets.

The results point towards random forest being the best overall performing algorithm. But when it comes to data with a lot of categorical dimensions the multi-layer perceptron was the top performer. And when it comes to data-manipulation, undersampling was the best approach for all the datasets and algorithms.

INNEHÅLLSFÖRTECKNING

1 Introduction

If a person were tasked with for example trying to guess whether a credit card transaction is fraudulent or not based on different transactions, the goal of the task is to get at least 80% of the guesses correct, whether they are a fraud or not doesn't matter. If this person then knew that just 1% of the transactions are in fact a fraud, then the logical approach would obviously be to guess that everything is not a fraud and thus achieve 99% correct guesses.

This is exactly what classification algorithms does if they are not correctly tuned based on the structure of the data. Naman et al. (2018) Describes that when general classifiers encounter imbalanced data, the algorithm favours the majority class in comparison to other classes. This leads to the classifier neglecting the minority class by trying to achieve the best accuracy, and this in turn leading to a skewed classification-accuracy drastically favoring the majority class. Meaning the accuracy is high but many or all instances of the minority class are misclassified.

This approach could of course be useful in some situations but imagine if the data that were being classified contains data about previously mentioned fraud cases as the minority class, or for example cancer patients. Then misclassifying the minority class could have enormous negative impact.

The demand of machine learning is constantly increasing in society as the amount of data collected by different organisations increase. And manually analysing data of this magnitude is becoming less and less feasible.

"The amount of data generated is increasing every day, this also increases the demand for learning systems which can predict, classify and analyse the data efficiently"
(Naman et al., 2018).

Therefore, this study aims to address the problem of classifying the minority class by benchmarking different algorithms, the algorithms are tested with different imbalanced datasets. Different methods for manipulating the data are also tested in combination with the datasets to ensure a thorough study of how to best tackle different datasets with different algorithms.

Examples of what a minority class can represent includes software defects (Rodriguez et al., 2014), natural disasters (Maalouf and Trafalis, 2011), cancer gene expressions (Yu et al., 2012), fraudulent credit card transactions (Panigrahi et al., 2009), and telecommunications fraud (Olszewski, 2012).

2 Background

This chapter starts of by addressing central terms and subjects, and how they are defined in the case of this study. Such as machine learning (*supervised, unsupervised*), class imbalance, classification, data manipulation and lastly how machine learning is applicable in information systems. Under these headings relevant literature from already existing theories regarding the subject are also presented.

2.1 Machine Learning

Machine learning (ML) can be defined as the answer to the question: *How can computers learn to solve problems without being explicitly programmed?* Koza et al. (1996). This can be achieved by only defining what problem the computer is tasked to solve, and not explicitly defining the path to solve said problem only the available resources (Koza et al., 1996). The computer achieves the objective by relying on pattern recognition and iteration through different types of algorithms. Since the process relies on pattern recognition the amount of data (Training Data) available for training is critical for the quality of the learning and how well the computer can learn the path to solve the problem. The practise of ML is applied in a variety of different scenarios, where it is infeasible or even impossible to develop a conventional system for performing the task.

"Machine learning is a highly interdisciplinary field which borrows and builds upon ideas from statistics, computer science, engineering, cognitive science, optimisation theory and many other disciplines of science and mathematics." (Ghahramani, 2003).

Below is a figure describing the workflow of machine learning (See figure: 1). Ingestion refers to the process of gathering data, i.e. the information that is going to be fed into the algorithm for training. Data prep is the step where the data is structured to suite the algorithm, unnecessary data and outliers are removed(*noise*). Then the algorithm iteratively trains using the input data until the performance doesn't improve. Then completely unseen data is introduced for prediction, the results are then evaluated to measure the performance.

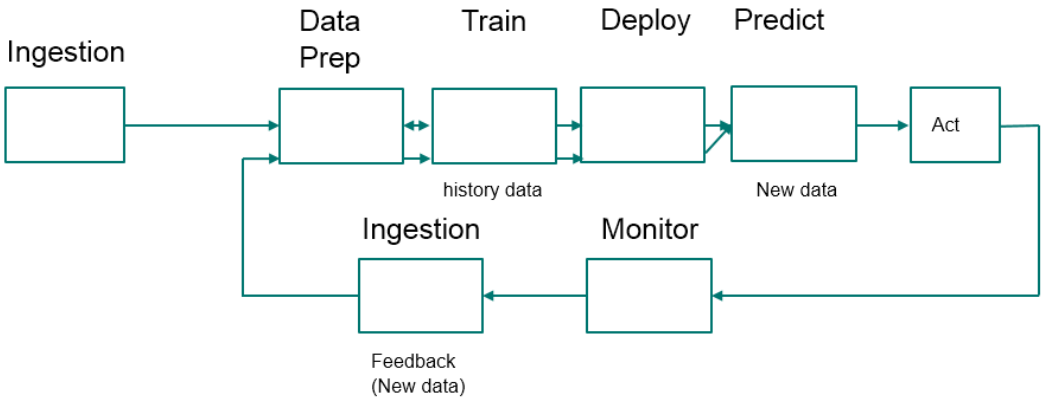

Figure: 1, Machine Learning Workflow

ML is considered a subset of Artificial Intelligence (AI), and consists of many different application forms, these are split into three major categories of tasks where ML is well suited: Supervised, Unsupervised and Reinforcement Learning. This study will focus on the supervised learning since classification that is the goal of the study falls under this category. Following sub chapters will go through unsupervised and supervised learning and disregard reinforcement learning since that category leans more towards AI and robotics than data analytics.

2.1.1 Supervised Learning

Supervised learning is a form of learning where you know what dimension of the dataset you are interested in, this meaning you know all the dimensions = *X,* and you know the target dimension = *Y*. And the goal of the training process is for the algorithm to learn to derive the value of *Y* by only knowing the values of *X.* This is done by iteratively training the algorithm by letting it try to predict *Y-values* only using the *X-values* and validating the results by comparing them to the already known *Y-values* and feeding the results back.
McCue (2014) describes the goal of supervised learning to be the development of rules or decisions for an algorithm based on data with known outcome. To then be able to introduce data with an unknown outcome and by the used of the previously determined rules or decisions, try to predict the outcome of the newly introduced data.
Supervised learning can be split into two categories: *Classification and Regression.* In the first mentioned classification the *Y-value* is representing a class like type of bird and the *X-values* are known facts about the different types of birds. Regression on the other hand serves to predict a fluctuating *Y-value,* the most common example is the *house price example: Y =* the price of a house and *X =* different features of the house like square-meters and number of rooms. And an algorithm is trained to predict the price of a house based on the square-meters and number of rooms. As previously mentioned, this study will focus on classification which is a subcategory of supervised learning.

2.1.2 Unsupervised Learning

Ghahramani (2003) explains unsupervised learning as methods that does not obtain *"supervised"* target outputs, nor rewards from its environment. It can be hard to grasp what the machine can achieve given no target output nor rewards/feedback, but the unsupervised learning is used to build representations of the given input. An example would be, combining different features into one such as BMI *(Body Mass Index)* which is a representation of your weight and height together in a combined metric. Unsupervised learning can be used to condense many dimensions into fewer ones. Another application of unsupervised learning is pattern recognition, this meaning recognizing complex patterns in big amounts of input data that would not be comprehensible by an ordinary person. Unsupervised learning can be split into two categories: *Dimension* reduction and *Clustering*. Since this study focuses on classification of imbalanced datasets the unsupervised algorithms are excluded to narrow the field, in other research they could be used as pre-processing before the classification.

2.2 Evaluation of Classification

Evaluating is a big part of the machine learning lifecycle, since it is the process where the performance of the model is measured and thus is one of the most critical steps for constructing a deployable product. Ranking classification algorithms normally need to examine multiple criteria, such as *accuracy, ROC-AUC, F1-measure* (Kou et al., 2012) *(see chapter: 4.4, for explanation of these metrics)*. To be able to calculate these different metrics a baseline needs to be set of how to score the classification algorithm. This is done by comparing the predicted value of the algorithm and the actual value of the predicted class. This leaves four different variables if the class-label is binary, in the case of this study the minority class is the target and proscribed as the *positive* and the majority class is then represented as *negative*.

The performance of the classification is the measured using four different outcomes of every prediction, which are the following: True Positive (TP, which represent all correct classifications of the minority class), True Negative (TN, which represent all correct classifications of the majority class), False Positive (FP, where a majority class has been classified as a minority class) and False Negative (FN, where a minority class was classified as a majority class). Figure: 2, below visualizes this with the actual class on the y-axis and the predicted class on the x-axis.

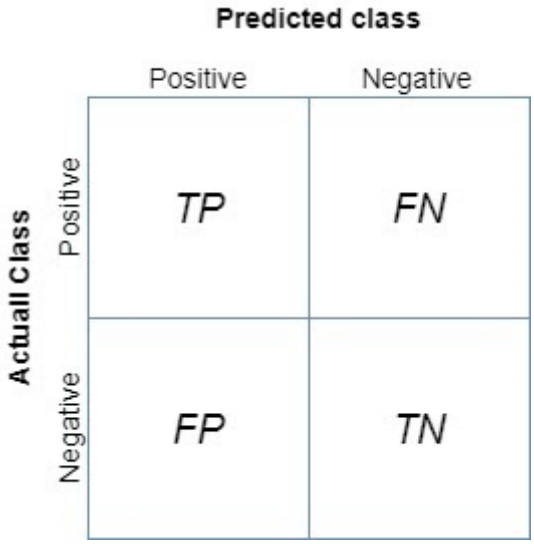

Figure: 2, Prediction outcome

2.3 Class Imbalance

Class imbalance in this study refer to a dataset containing observations of two or more different classes with imbalanced distribution. Chawla et al. (2003) describes this imbalance as datasets with a very infrequent occurrence of the minority class, with a frequency ranging from 5% to less than 0.1%. As the amount of minority class observations are low, the magnitude of these observations can still greatly outweigh the majority class in terms of impact.

King & Zeng (2001) tries to define the minority class, they explain this phenomenon as a binary variable that *is* or *is not* a minority class observation with dozens to a thousand less minority classes than majority classes. This seems to be quite a broad statement and does not serve the purpose of narrowing down the definition. Even so it can provide somewhat of a baseline for this study, where the middle ground of their definition is selected as a baseline for determining what is considered an imbalanced dataset. For this study, a minority class is considered an observation that occurs 1% or less times in comparison to all observations. To determine if the data has the preferred distribution, the amount of minority class observations is divided by the total number of observations and if the product of that equation is =*<0.01* it will be defined as an imbalanced dataset.

2.4 Data-Manipulation

Data-manipulation refers to the process of transforming data into something, hopefully more useful for training the machine learning algorithms. This process consists of many different approaches such as synthetically generating new data to reinforce the minority-class *(up-sampling or augmenting)*. Another approach is to draw a stratified sample of the majority-class that is closer in size to the minority-class *(undersampling)* thus balancing out the datasets and countering biases. Zhiting et al. (2019) suggest that weighting the data according to its importance usually is a more rigorous approach, because by weighting no synthetic data is created, or any actual data is removed thus allowing the dataset to stay as reflective as possible of what was originally recorded.

2.5 Classification of Imbalanced Data

To be able to precisely and efficiently locate the minority class with machine learning, all points made above *(headings: 2.1 - 2.3)* can have significant impact on the results or the rigor of said results. Li et al. (2016) also claim that all data is different and thus should be handled differently for the results to be optimal. They also claim that applying a single specific ensemble classifier to tackle different kinds of imbalanced-data was an inefficient approach, because such a classifier would need to be tuned accordingly to the data iteratively thus making a *one for all solution like that very inefficient* (Li et al., 2016).

However, Haixiang et al. (2017) suggest the use of ensemble-based algorithms to account for biases caused by imbalanced datasets. This goes in line with what Zhiting et al. (2019) points at about trying to tune the algorithms instead of manipulating the data *(see chapter: 2.3),* by employing a ensemble method many so called *"weak learners"* (*underperforming algorithms*) work together by the use of a majority vote system thus accounting for local biases had by individual learners.

2.6 ML in Information Systems

This chapter serves to ground the research in the subject of information systems, how machine learning fits into the broad field of information systems. And to further motivate how the machine learning task of classification can create new and improved methods for analysing data, and thus supporting the flow of information in organizations. Bose & Mahapatra (2001) describes how data warehousing technology has allowed companies to store large volumes of data in an organized fashion. They further explain that the sheer volume of data in organizations

have then in turn given rise to the task of data mining, by automating tedious but important tasks that would be very time consuming or even impossible to perform manually. Data mining is described as following:

> *"Modern data mining combines statistics with ideas, tools and methods from computer science, machine learning, database technology and other classical data analytical technologies." (Hand, 2007).*

Machine learning techniques are used for analysis and pattern discovery and thus play a big role in the development of data mining applications as stated by Bose & Mahapatra (2001). As previously mentioned, there are many different machine learning techniques which each comes with its own strengths and weaknesses. Bose & Mahapatra (2001) pushes the importance of information system managers understanding these different techniques, to be able to correctly incorporate them in data mining tools.

2.6.1 Classification in Information Systems

As previously mentioned, classification is a subset of techniques from the field of ML. This subchapter aims to explain how classification fits into the information systems of organizations.

Amani & Fadlalla (2017) conducted a study of publications regarding data mining applications in the finance sector. Their analysis reveals that the classification task represents a vast majority (67%) of data mining applications.

A wider study regarding business in general was conducted by Bose & Mahapatra (2001), they also found classification to be the dominant problem category (31.7%) of all applications. They further explain the financial sector to be the predominant sector in adopting machine learning, with tasks such as: categorizing the risk-return characteristics of stocks, bonds and mutual funds, and determining the creditworthiness of a credit application. The statements above points towards the relevance of classification within the field of information systems with different types of organizations.

3 Problem Area

This chapter serves to clarify exactly what challenges this study tries to tackle. It starts of by exploring related studies by explaining how these contributed to the research and how this research differs from previous work. Afterwards the aim of the research and what gap the research tries to fill is defined. Followed by a motivation as to why the research is important, and how the results can contribute to the field. Lastly the research question is formulated through a condensation of what the research incorporates.

3.1 Related Research

Caruana & Niculescu-Mizil (2006) prescribes the vast amount of proposed supervised learning algorithms and performs a comparative study of ten different algorithms over eight evaluation metrics. They found boosted trees to be the best performer overall, closely followed by random forest. Huang et al. (2016) investigates data manipulation in deep learning with class imbalanced data of three different datasets, they found undersampling to be the most efficient approach, and oversampling to be inefficient.
Haixiang et al. (2017) performed a comprehensive literature study of five hundred and seventeen papers related to supervised learning for imbalanced datasets. They big diversity in proposed methods regarding classification algorithms, data pre-processing and model evaluation. Datta & Arputharaj (2018) compared several methodologies for class imbalance through a literature review, they compare decision trees, association mining and ensemble learning over 20 different datasets.

Drummond & Holte (2003) investigated the interaction of undersampling and oversampling with decision tree algorithm C4.5. Which is still a well-adapted model although most implemented in ensemble learners. They found undersampling to account well for biases and oversampling to be surprisingly inefficient.

Kou et al. (2012) proposes the weighting algorithm MCDM, through and experimental study in which they apply seventeen classifiers over eleven different datasets. Tan et al. (2019) show improving results with algorithms for data augmentation and weighting. They propose that further research is done into optimization by combining different data manipulation schemes.

All the different studies above try to solve the broad problem of class imbalance within the field of machine learning. By quite different approaches to improve on the existing methodologies within the field, either by proposing new algorithms and or comparing existing ones.

3.2 Aim

To tackle the problem of class imbalance a vast amount of methods is presented (Chapter: 3.1) ranging from different data manipulations methods with different types of classification algorithms. But the gap this research tries to fill is the lack of a broader comparison of the combination of both classification algorithms and data manipulation methods on different types of imbalanced data. The aim of this study shares similarities with the research performed by Huang et al. (2016) where they try different algorithms with different data manipulation on three datasets, however all datasets they used consisted of images and thus limits the research results to image classification. This study on the other hand uses three differently shaped tables of data regarding: companies, credit card transactions and individuals.

So, the aim is to benchmark different algorithms, in combination with a variety of data manipulation methods to classify different types of imbalanced data, to help individuals within the field or organizations to easier find the correct combination for their data. This will be done through comparative experiments using three commonly applied classification algorithms: Random Forest (RF), Gradient Boosted Decision Trees (GBDT) and Multi-Layer Perceptron (MLP) in combination with three different approaches to data manipulation: Oversampling, undersampling and no data-manipulation.

3.3 Motivation

Classification is one of the most important machine learning tasks. It is a method that is applicable in many different fields and can predict the labels of new unknown data making them a powerful resource for many different types of organizations. However, most original classification algorithms pursue the highest possible accuracy by minimizing the error rate (*the amount of incorrect prediction of class labels*). They tend to assume that all misclassification errors have the same impact (Ling & Sheng, 2008). Now imagine this being the case when trying to classify for example: Fraud, cancer patients or seismic activity. Where of course a misclassified non-occurred event of fraud case or seismic activity *(False positive, see chapter: 2.2)* would cause some disturbance, the point to be made is on the other hand far worse, where a truly occurred fraud or a patient with cancer are misclassified as non-fraudulent or not being sick.

Accuracy is a commonly used metric for evaluating the performance of machine learning algorithms. Although it reflects the error-rate it does not account for the weight *(impact)* of misclassifying different classes. And for imbalanced data sets, this performance measure may not mean the best since it is always biased to the majority class according to Li and Wong (2015).

> *"Two fundamental assumptions are often made in the traditional cost-insensitive classifiers. The first is that the goal of the classifiers is to maximize the accuracy (or minimize the error rate) the second is that the class distribution of the training and test datasets is the same."* (Provost, 2000).

To further motivate the possible contributions of the study, let us say a researcher is presented with the task of classifying a dataset with significant class imbalance. The amount of suggested methods is vast, and some methods are better than others depending on the structure of the data. And as stated by Li et al. (2016) all data is different and should therefore be handled differently to optimize the results.

Bose & Mahapatra (2001) describes that information systems managers are often overwhelmed with the plethora of ML techniques, which further motivates the relevance of a study regarding which combination of methods suits what type of data. Coming researches can compare the data to the data used in this study and if it's structurally similar to any of the datasets used, then the best performing data manipulation methods and algorithms found in this study should be well suited for the new dataset. By conducting a benchmarking study like this, it can help coming research to faster find a well suited combination of data manipulation methods and algorithms for their data. Hence it is important to have variation between the data sets, to be able to cover a wider range of data structures.

3.4 Research Question

This study aims to find the optimal combination of classification algorithms (RF, GBDT, MLP) and data manipulation methods (oversampling, undersampling and no data manipulation), for three datasets of different shapes with different characteristics. And then determine which combination is best suited for which type of data. In comparison to previous studies, the goal is to create an overview of previously adopted methods in combination and then compare these to each other. As previous studies either compare different ML-models or different data manipulation methods, but not the two in combination. And to further narrow the scope, the study focuses on datasets with unequal distribution of classes. This boils down to the following research question:

Which combination of chosen classification algorithms and data manipulation methods have the best performance for each dataset?

3.5 Limitations

As this study tries to compare and benchmark different algorithms, emphasis is put on the equality of optimization between the different implementations. This means the models might not be optimally tuned but tuned to and equal extent to make the results comparable.

The study does not go into the inner workings and the mathematics within the algorithms. This because the goal is not to propose a new approach or optimize a certain algorithm. But rather make a clear comparison between well adapted models within the field. Hence the inner workings of the algorithms will to some degree be treated as black boxes.

The subject of the machine learnings performance from a material and computational standpoint, such as run time and leverage over computational hardware has a big role in the field. This is acknowledged as an important subject in the field, but for the scope of this study is excluded. Meaning the run-time and efficiency of the algorithms will not be measured, optimized or evaluated.

4 Research Method

This chapter defines the course of action the research will take to ensure the rigor and validity of the research. It will define how *data, data manipulation methods, machine learning algorithms, evaluation methods and algorithm optimization* are selected and/or implemented. The research will be conducted in a purely experimental manner, meaning experiments will be conducted evaluated and documented. Said experiments will be consist of test using following machine learning models: *Random forest (RF), Gradient Boosted Decision Trees (GBDT) and Multilayer Perceptron (MLP)*. In combination with following data manipulation methods: *Undersampling, oversampling and no data manipulation.* The results of the experiments will then be evaluated *(see chapter: 4.4 Evaluation)*, documented and compared to each other for analysis. For a visualization of how the experiments will be structured see: figure: 3, experiment structure.

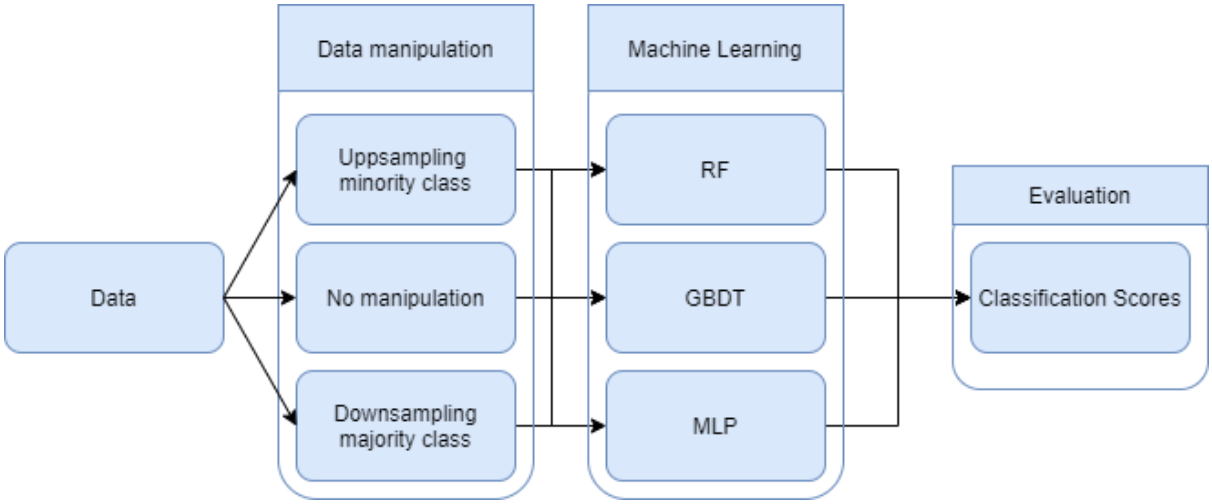

Figure: 3, experiment structure

4.1 Data

The data source for this research will be open free to use datasets accessible online.
However, since the goal of the research is focused on classification of imbalanced datasets two criterion for the datasets will be mandatory, assuring that the data is relevant for the purpose of the research. The first one is that the dataset contain enough samples *(n)* and dimensions *(d)* for machine learning to be applicable to it. The dataset must have the following shape:

$$n => 1\,000 \text{ and } d => 10$$

The second criteria are that the data must reflect some sort of class imbalance following the definition made in *chapter: 2.3.* Meaning it must consist of at least two classes, having a majority class that is at least one hundred times bigger than the minority class.
It is also important for the datasets to have structural differences, such as size, class distribution and data recorded in the columns. Hence allowing the experiments to represent or have similarities with as many real-world datasets as possible, as the goal of the study is to serve as a road map in choosing the correct data manipulation method and classification algorithm for feature research.

4.1.1 Datasets

Below the datasets that is used in the study are presented together with a motivation as to why they are well suited for the study, other than the predetermined criterion presented in *chapter: 4.1.*

Dataset: 1, Financial Distress.

The first dataset consists of 3672 rows and 86 columns, it represents the financial status of several companies. The columns represent different facts about the status of the company, such as time of the observation, credit score, number of employees and the target class, that determines whether the company is in financial distress or not. All the data is numerical and normalized and contain no missing values (*null*). The amount of observations representing a company in financial distress is 136, making the fraction of target observations: 0.038. See figure: 4, dataset 1 head, for a example of the first rows and columns of the dataset.

This dataset is very well adjusted of machine learning because it only contains numeric values, contains no missing values and all the values are normalized. It is also a so called *labled* dataset, meaning the target variable is assigned known for every observation making it a good examples dataset for this study. It also shares the type of variables and values with many real-world datasets.

	Company	Time	Financial Distress	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13
0	1	1	0.010636	1.2810	0.022934	0.87454	1.21640	0.060940	0.188270	0.52510	0.018854	0.182790	0.006449	0.85822	2.00580	0.125460
1	1	2	-0.455970	1.2700	0.006454	0.82067	1.00490	-0.014080	0.181040	0.62288	0.006423	0.035991	0.001795	0.85152	-0.48644	0.179330
2	1	3	-0.325390	1.0529	-0.059379	0.92242	0.72926	0.020476	0.044865	0.43292	-0.081423	-0.765400	-0.054324	0.89314	0.41220	0.077578
3	1	4	-0.566570	1.1131	-0.015229	0.85888	0.80974	0.076037	0.091033	0.67546	-0.018807	-0.107910	-0.065316	0.89581	0.99490	0.141120
4	2	1	1.357300	1.0623	0.107020	0.81460	0.83593	0.199960	0.047800	0.74200	0.128030	0.577250	0.094075	0.81549	3.01470	0.185400

figure: 4, dataset 1 head

Dataset: 2, Credit card Fraud.

This dataset consists of 284315 rows and 31 columns, it represents different credit card transactions, with the columns representing time, place amount withdrawn etc. The target variable in this dataset is whether the transaction was fraudulent or not. This dataset also consists of numerical normalized data. The amount of fraudulent observations is 492, and thus the target variables fraction: 0.0017. See figure: 5, dataset 2 head, for a example of the first rows and columns of the dataset.

The dataset shares some similarities with *dataset 1* but the big difference is the size of it and the much smaller fraction of target observations. This leads to it being a good dataset to the smaller one in terms of benchmarking the different algorithms.

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	0.090794	-0.551600	-0.617801	-0.991390	-0.311169
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	-0.166974	1.612727	1.065235	0.489095	-0.143772
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	0.207643	0.624501	0.066084	0.717293	-0.165946
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	-0.054952	-0.226487	0.178228	0.507757	-0.287924
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	0.753074	-0.822843	0.538196	1.345852	-1.119670

figure: 5, dataset 2 head

Dataset: 3, Customers.
The last dataset has 1723 rows and 14 columns,the rows represent different customers of a company, the columns represent facts about the individual customers. This dataset contains a mix of both categorical values and numerical values. The target variables in this dataset is whether a customer is good or bad for the company. The amount of observed bad customers is 196 making the fraction of the target variable: 0.128. See figure: 6, dataset 3 head for an example of the first rows and columns of the dataset.

This dataset was chosen as a comparison to the two purely numerical datasets. Since performing machine learning with categorical values varies from using only numerical data, making it a good dataset for showing which algorithm performs better with categorical columns.

	month	credit_amount	credit_term	age	sex	education	product_type	having_children_flg	region	income	family_status	phone_operator	is_client
22	1	28000	18	36	male	Secondary special education	Cell phones	1	2	16000	Another	2	0
23	1	14000	10	25	female	Secondary special education	Cell phones	0	2	24000	Another	1	1
57	1	8500	6	28	male	Secondary special education	Cell phones	0	2	19000	Unmarried	3	0
61	1	8000	12	25	female	Secondary education	Household appliances	0	2	11000	Married	3	1
64	1	15500	10	25	female	Secondary education	Household appliances	0	2	19000	Another	1	1

figure: 6, dataset 3 head

4.2 Data-manipulation Methods

Tsai et al. (2019) describes it to be exceedingly difficult to construct effective classifiers, especially for distinguishing the minority class. They describe the most commonly used method to tackle this problem is the under/oversampling approach, where either the majority class is reduced *(Undersampling)* or the minority class is reinforced with synthetic data *(Oversampling),* see: Figure: 3, experiment structure, for a visualization of how these methods fit into the experiments.

Oversampling
Oversampling is a technique widely used in the field of machine learning. The goal is to even out the distribution of observations per class in a dataset, this is done by synthetically generating data of the minority class based on the already existing instances. Zhu et al. (2019) describes it as the most common solution for class imbalance problems, furthermore they point towards the risk of incorrectly generating the synthetic data, which will result in the dataset no longer representing the real world observation, thus making it un-useful in training a deployable machine learning model.

Huang et al. (2016) explains that oversampling in many cases tend to overfit the model, which means the model fails to generalise between the different classes and rather just learns to locate specific data points. And thus, failing to increase the amount of useful information in the data. This in turn leads to low performance on data the model has not seen before since it's just trained to single out data points and not generalise.

The method for oversampling that will be applied in this study is ADASYN proposed by He et al. (2008), they claim that ADASYN can adaptively generate synthetic data samples for the minority class to reduce the bias introduced by the imbalanced data distribution.

"The essential idea of ADASYN is to use a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn." (He et al., 2008)

Undersampling

Huang, C. et al. (2016) claims that under-sampling often is the preferred course of action. In comparison to the previously mentioned oversampling, undersampling rather removes observations from the majority class until the classes are equal in size. Huang, C. et al. (2016) further advise the use of undersampling based on the performance tests in their study, even tho the potential loss of valuable information the undersampled data outperformed the oversampled set.

Drummond and Holte. (2003) conducted a study where they compare over and undersampling together with a tree-based algorithm. They found undersampling to have a reasonable sensitivity to changes in misclassification. Furthermore, they found oversampling to often produce little or no change in the performance. These results are relevant because the model *random forest (see chapter: 4.3 Machine learning models)* which is implemented in this study is based on the original tree algorithm they implement.

No data-manipulation

Tsai, C. F. et al. (2019) also suggests a third approach, where the data is not manipulated before the application of the algorithms. And the algorithms are instead tuned to prioritize correctly classifying specific classes (Cost-sensitive learning), this is done by implementing heavier penalty for wrongly classified minority classes. Huang, C. et al. (2016) suggests this to be the most efficient method because it reduces the amount of pre-processing and can guarantee the same performance with different batches of data. Huang, C. et al. (2016) also points to the benefits of not pre-processing the data and thus minimizing loss of data and ensures the data represents the original observations.

To conclude this chapter, as described above there are a lot of studies in favour of each method. This study will therefore test all three approaches to tackle imbalance in the datasets, and to try to set a baseline for which method works best with what type of data and classification algorithm. Please see: Figure: 3, experiment structure, for a visualization of how these methods fit into the experiments and Figure: 7, resampling, for an example of undersampling and oversampling.

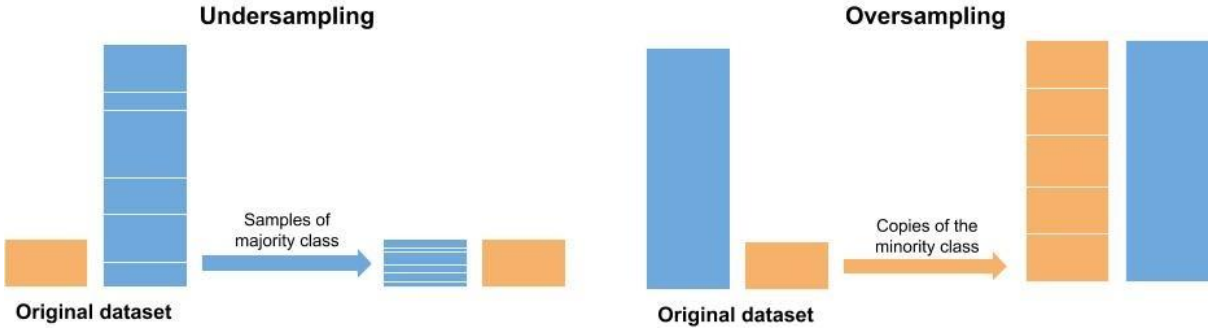

figure: 7, resampling

4.3 Machine Learning Models

This chapter serves to motivate the different models that were chosen and briefly explain how said model works. Note that the descriptions are general explanations of the theory behind the algorithms and not the mathematical inner workings of the algorithms, since it is not in line with the goal of the research to go in depth into how they operate.

4.3.1 Random Forest (RF)

Biau & Scornet (2016) describes RF (random forest) to be an extremely successful general-purpose classification method due to its ability successfully handle a large amount of variables. The model is further motivated by claim made by Belgiu & Drăguţ (2016) which states it to be both fast and insensitive to overfitting. Lessmann et al. (2015) conducted a study to benchmark classification algorithms, and the RF was one of the top performers in the majority of the tests.

Zhang, C. et al. (2017) also shows RF to be a top performer in the field of classification, with their study were they benchmark the algorithm across 71 different datasets, where RF had the best total average accuracy score of all models tested. These results make RF a natural choice for this study as it is one of the state-of-the-art classifiers.

RF is a parallel ensemble learning model, that means the model, consists of *n* amount of individual decision trees. Then a majority vote approach is used, meaning the majority of the best performing trees are used to cast a vote. It operates according to the "divide and conquer" approach by sampling fractions of the data in the individual trees according to Biau & Scornet (2016).

4.3.2 Gradient Boosted Decision Trees (GBDT)

Si et al. (2017) Describes GBDT (Gradient Boosted Decision Trees) as a powerful machine learning technique that has a wide application range in both academic and commercial use and produces state of the art results in many different data mining challenges. In the benchmarking study by Zhang et al. (2017) GBDT was a also a top performer together with RF in terms of accuracy. Besides the accuracy performance Saberian et al. (2019) describes GBDT to have a small memory footprint by the construction of smaller trees, and as all the tree-based approaches a decent amount of explainability in the results.

As described by Ravanshad (2018) GBDT in comparison to RF build trees one at a time, where each new tree helps to correct errors made by previously trained tree. So instead of creating parallel trees, and then using the majority vote GBDT feeds information from each tree to the next.

4.3.3 Multi-Layer Perceptron (MLP)

Zanaty (2012) describes the MLP (Multi-Layer Perceptron) as perhaps the most popular network architecture in use for both classification and regression. Note that their study dates to 2012 which is a considerable time in this rapidly evolving field. Even though, the algorithm is still implemented in many different settings. Hence making it a good choice for and implementation of neural networks in this study.

MLP's are feedforward neural networks which are typically composed of several layers of nodes with unidirectional connections, often trained by back propagation (Zanaty, 2012). And as suggested by Foody (2004) feedforward neural networks are desirable for supervised classification.

The training consists of a technique called backpropagation; it's bidirectional meaning signals are passed both ways in the network. A training vector is sent forward through the network and is classified. And signals traveling backwards are gradients calculated of the loss function in the network. The output of the loss function is the value representing errors or in this case misclassifications. The backwards traveling signals then updates the weights of the layers as the model iteratively trains.

4.4 Evaluation

This is one of the most crucial steps in the process, for this study to maintain research rigor towards the validity of the results the evaluation must follow a organized and replicable structure. Below every metric used for evaluating the classification models are described.

"In the imbalanced classification domains, ROC (Receiver Operating Characteristics) are considered the "gold standard" of a classifier's ability. However, using only the ROC to select a potentially optimal classifier is not enough. In fact, the ROC curve and the AUC values reflect only the ranking power of positive prediction probability." (Zou et al., 2016).

ROC is based on probability and calculates a baseline for random guesses of the classes and evaluate the model based on how much better than random guesses it performs (Fawcett, 2006).

The models will be evaluated based on five different metrics: F1-score, accuracy, precision, recall and ROC. This chapter serves to explain and motivate these metrics. Below there are four equations explaining the four first mentioned evaluation metrics. All the evaluation builds upon the previously mentioned True Positive (TP, which represent all correct classifications of the minority class), True Negative (TN, which represent all correct classifications of the majority class), False Positive (FP, where a majority class has been classified as a minority class) and False Negative (FN, where a minority class was classified as a majority class).

$$Accuracy = (TP + TN) \div (TP + FP + TN + FN)$$
$$Precision = TP \div (TP + FP)$$
$$Recall = TP \div (TP + FN)$$
$$F1 = 2 * ((Precision * Recall) \div (Precision + Recall))$$

The fifth evaluation metric is ROC-curve. This metric will be useful where no data manipulation is implemented since it accounts for the class imbalance when scoring the performance of the model. The diagonal blue line (y = x) in figure: 8, ROC-Curve example, represents randomly guessing a class, thus creating a baseline independent of the class distribution. If a classifier randomly guesses the positive class half the time, it can be expected to get half the positives and half the negatives correct; this yields the point (0.5, 0.5) in ROC space. If it guesses the positive class 90% of the time, it can be expected to get 90% of the positives correct but its false positive rate will increase to 90% as well, yielding (0.9, 0.9) in ROC space (Fawcett, 2006). Now see the red line in figure: 8, ROC-Curve example, which represents the performance of the classifier, and the gap between the blue and red lines represents how much better than random guessing the classifier performs.

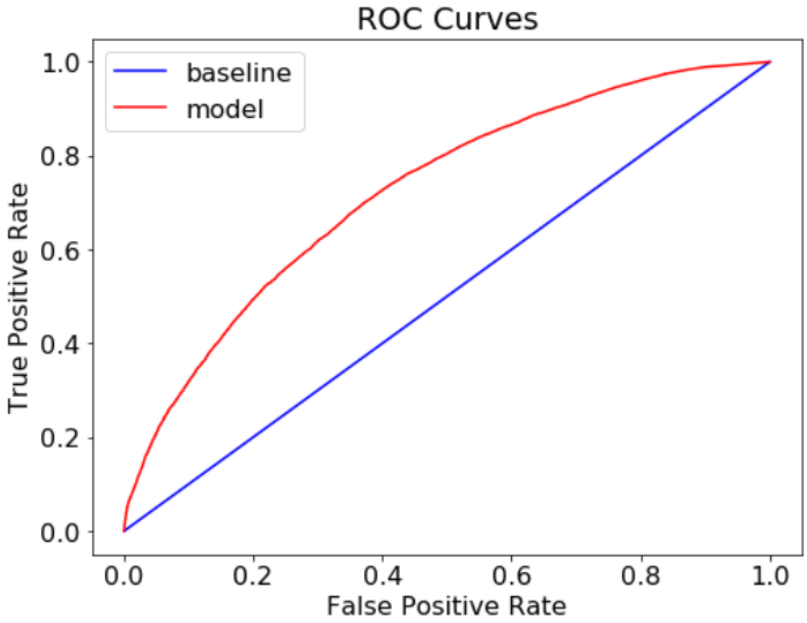

Figure: 8, ROC-Curve example

Fawcett (2006) further explains ROC graphs to be based upon TP-rate and FP-rate, in which each dimension is a strict columnar ratio, hence making it independent of class distribution in comparison to other classification evaluation metrics.

4.5 Model Optimization

To optimize the performance of the algorithms different hyperparameters needs to be tuned. These can for example be the number of training iterations the algorithms dose or how deep and wide a decision tree can be.

Claesen & De Moor, 2015 describes model optimization as training a model which minimizes a predefined loss function such as mean squared error or error rate. By tuning the hyperparameters of said model, these can be viewed as a sort of restrictions to optimize the performance.

"The choice of hyperparameters can significantly affect the resulting model's performance, but determining good values can be complex; hence a disciplined, theoretically sound search strategy is essential." (Claesen & De Moor, 2015)

To tune hyperparameters there are two different approaches, grid search and randomized search. In grid search a grid of all possible combinations of different hyperparameter values are set up. For each combination, the model is re-trained and scored on the test set. Worchester (2019) describes this as a thorough but inefficient method of tuning the hyperparameters and gives following example: Searching 20 different parameter values for each of 4 parameters will require 160,000 trials of cross-validation. IN comparison, random search sets up a grid of hyperparameter values and instead just select random combinations, this allows the number of search iterations is set based on time or resources. Worchester, P. (2019)

Bergstra & Bengio (2012) suggests the random search approach is the most efficient method and recommend random search over grid search to optimize runtime and use of computational resources. They make a strong case for random search over grid search, by performing tests with several different algorithms over different datasets and claim it to be more efficient because not all hyperparameters are equally important to tune. Since the project was limited by computational power, this approach was the best choice based on computation time contra results.

5 Results

The results of all the experiments are presented through four different tables, three tables for every data manipulation method, followed by a table presenting the best combination of data-manipulation method and algorithm per dataset. The first three tables show all the evaluation metrics per model and dataset. Note that all the top scoring results per metric and dataset are highlighted with a bold font. Also the amount of TP and FP are presented at the end of the tables except for the table of best performing combination.

Following table: 1, Undersampling results. Presents all the classification scores where the data sets were undersampled. These results are shown first because this method of data manipulation was the most consistent and best performing over all the experiments. Here we can see that the RF-model slightly outperforms the other models when it comes to ROC_AUC. An interesting note is MLP on dataset 3, which showed great precision, meaning it was successful in locating the minority target class, however the low overall scores suggest misclassification on the majority class.

Undersampled Data		ROC_AUC	F1	Precision	Recall	#TP	#FP
	RF	**0.91**	**0.91**	**0.90**	0.93	**37**	4
Dataset: 1	GBDT	0.90	0.90	0.85	**0.95**	35	6
	MLP	0.68	0.69	0.71	0.67	29	**12**
	RF	**0.92**	0.92	**0.93**	0.91	**138**	10
Dataset: 2	GBDT	**0.92**	**0.93**	**0.93**	0.93	137	11
	MLP	0.76	0.24	0.14	**1.00**	20	**128**
	RF	**0.66**	**0.67**	0.66	0.67	39	20
Dataset: 3	GBDT	0.65	0.62	0.58	**0.68**	34	**25**
	MLP	0.61	0.67	**0.80**	0.57	**47**	12

Table: 1, Undersampling results

Following table: 2. Shows all the scores from the algorithms in conjunction with oversampled datasets. The scores are considerably worse than undersampling, but a big upside to this method is that no original data is removed. Here we notice that MLP overfitted on both dataset two and three, which could arguably be remedied by some hyperparameter tuning. But to maintain the rigor of equally implementing and optimizing the algorithms, the results are presented as such.

Oversampeld data		ROC_AUC	F1	Precision	Recall	#TP	#FP
	RF	0.59	0.30	0.54	0.21	22	19
Dataset: 1	GBDT	**0.61**	0.30	0.34	**0.26**	14	**27**
	MLP	0.60	**0.33**	**0.61**	0.23	**25**	16
	RF	0.66	0.46	0.82	0.32	121	27
Dataset: 2	GBDT	**0.78**	**0.66**	0.76	0.58	113	**35**
	MLP	0.49	0.00	**0.99**	0.00	**146**	2
	RF	**0.58**	**0.27**	**0.29**	**0.26**	**17**	42
Dataset: 3	GBDT	0.55	0.22	0.25	0.20	15	44
	MLP	0.50	0.00	0.00	0.00	0	**59**

Table: 2, Oversampling Results

Furthermore, the results of no data manipulation are shown below in *Table: 3.* Here exceptionally low number of true positives for dataset: 1 in comparison to the other methods. Also, the tp-rate for dataset: 3 was incredibly low except for GBDT which performed surprisingly well.

No data-manipulation		ROC_AUC	F1	Precision	Recall	#TP	#FP
	RF	**0.70**	0.16	0.10	**0.44**	4	**37**
Dataset: 1	GBDT	0.63	0.28	0.27	0.29	11	30
	MLP	0.69	**0.36**	**0.32**	0.41	**13**	28
	RF	**0.96**	**0.85**	0.78	**0.93**	116	32
Dataset: 2	GBDT	0.65	0.45	**0.80**	0.32	**118**	13
	MLP	0.49	0	0	0	0	**148**
	RF	0.55	0.06	0.03	0.22	2	57
Dataset: 3	GBDT	**0.60**	**0.33**	**0.37**	**0.29**	**22**	37
	MLP	0.49	0.00	0.00	0.00	0	**59**

Table: 3, No data-manipulation results

To conclude the results the following two tables Table: 4, and Table: 5. The first one shows the best combination of data manipulation and algorithm per dataset and the second an average of all the scores for every data manipulation method.

	Algorithm	Datamanipulation methood	ROC_AUC	F1	Precision	Recall
Dataset: 1	RF	Undersampled	0.91	0.91	0.93	0.93
Dataset: 2	RF	Undersampled	0.92	0.92	0.93	0.91
Dataset: 3	MLP	Undersampled	0.61	0.67	0.80	0.57

Table: 4, Best performance per dataset

	Avg ROC_AUC	Avg F1	Avg Precision	Avg Recall
No data manipulation	0.64	0.27	0.29	0.32
Undersampling	0.77	0.78	0.72	0.81
Oversampling	0.60	0.28	0.51	0.22

Table: 5, avg of data-manipulation methods

6 Analysis

This chapter serves to further investigate what the results actually tells us. This is done in a structural manner, where each of the experiment components are analysed individually. First of the data manipulation methods: Oversampling, undersampling and no data manipulation. Followed by the classification algorithms: Random Forest, Gradient Boosted Decision Trees and Multi-layer perceptron.

Oversampling
Looking at the averages in *Table: 5, avg of data manipulation methods* oversampling actually didn't contribute at all to the overall performance of the modells, and in many cases instead decreased performance and seemed to add more noise to the data. But in terms of locating the target class performance slightly increases for all the algorithms and data sets. This method increased the ability to locate the target class through sacrificing overall performance, meaning a higher amount of TP and FN. This goes in line with what Huang et al. (2016) found about oversampling tends to overfit the model which in turn leads to the model failing to generalise between the classes.

Undersampling
This was by far the best overall method for data manipulation, in comparison to no data manipulation. Undersampling increased the overall scores for all algorithms on all of the datasets except for RF on dataset: 2 which actually decreased in overall performance, but with an increased TP-amount which is the most valuable of the metrics.

The results go in line what previous studies have found, Huang et al. (2016) suggests undersampling and show similar results in their study with undersampling being the best performing method.

No data manipulation
Overall performance wise no data manipulation scored slightly higher than oversampling. But all the algorithms had relatively low TP-rates. And it seems the performance was inherently based on the ability to locate the majority classes, and not the targeted minority class. Thus, making it the worst method for location the target class, but slightly better when the goal is to differentiate between classes.

It's worthy to mention that there is no data loss when this data manipulation method is used, this allows the data properly represent the original observations. This might be valuable in some applications.

Random Forest (RF)
Was the best performing algorithm in combination with undersampling which was the best data manipulation method. When implemented with the other data-manipulation methods it was performing equally with GBDT. Biau & Scornet (2016) describes random forest to be an extremely successful general-purpose classification method due to its ability successfully handle many variables, which this study's results also point towards.

It is also interesting to look at the TP-rate for RF when combined with the other two data manipulation methods, undersampling and no data manipulation. In this case RF is always outperformed by the other two algorithms.

Gradient Boosted Decision Trees (GBDT)

BGDT was a type of average performer in majority of the experiments, it was rarely the top performer but also not the worst performer. It proved to be a very versatile algorithm, a parallel could be drawn to a swiss army knife where it is a tool with many possible applications but rarely the best tool for the job.

If one algorithm were to contest the claims of Li et al. (2016) about a single ensemble classifier being very inefficient in tackling different kinds of imbalanced data, GBDT would be that algorithm based on the results of this study, as it proves to be very adaptable.

Multi-layer perceptron (MLP)
The most interesting results for the MLP algorithm was with the undersampling data manipulation, and on dataset: 3. In this case the MLP was successful in locating a lot of the minority class. Considering dataset 3 was the hardest one to classify for all of the algorithms this is interesting, the dataset contains a lot of categorical columns which causes the dataset to contain many variables, as every categorical value in a column has to be flipped into its own binary column.

When it comes to the other data manipulation methods, *overampling* and *no data-manipulation* MLP did not perform well. As we can see from the classification-scores the modell was overfitting, this is probably caused by the training datasets being too large for the MLP to handle. The results go in line with what Huang et al. (2016) explains, that oversampling in many cases have a tendency to overfit the model, which means the model fails to generalise between the different classes and rather just learns to locate specific data points. This could probably be prevented by fine tuning the hyperparameters more, but a decision was made to treat the tuning of the models the same way and keep it even between them. As a keenness to overfit with a larger amount of data is a result in its own for this study, as the goal is to evaluate the different models.

7 Discussion and Conclusion

7.1 Validity

It is of great importance to correctly evaluate the performance of the models, this study took advantage of four different metrics to evaluate the results. This leads to a clear picture of the performance of the algorithms, by showcasing all aspects of how well each individual model was differentiating between classes.

As for the optimization of the different algorithms by hyperparameter tuning with randomized search, two different points can be made around the validity of the experiments. First of this maintains an equality between how much tuning was conducted for each model, leading to rigorous results in performance in the aspect of how much tuning was required for the model to work. The other side of the coin is the case of MLP which was obviously overfitting with this amount of tuning, thus not correctly showcasing the models true potential in the results with oversampling data and unmanipulated data.

Only ADASYN was implemented for oversampling the data, as it is a powerful algorithm that correctly replicate the variance and distribution of the minority class He et al. (2008). Maybe some other methods for oversampling could have been tested to further reinforce the validity of the results.

7.2 Societal Aspects

As previously mentioned, the minority classes in different dataset can represent highly impactful elements in society. The minority class can in some cases take the shape of, for example diseases, criminal activity or even natural disasters. And to be able to efficiently and correctly locate these cases within a dataset can bear great importance for society. As this study contributes to this goal by trying to benchmark different algorithms in relation to different kinds of datasets with varying structure, to be able to cover as much ground as possible.

7.3 Scientific Aspects

The study does not propose any new methods or ways to configure existing ones, but rather tries to benchmark combinations well adapted and accepted techniques. This is also important work to evaluate what model works best with which type of data and data manipulation method, and document how well the different models perform in different settings. The study can contribute by acting as a form of guidelines when choosing a model and/or data manipulation technique for a specific type of data. This can speed up the process when presented with a new dataset.

By treating the algorithms as black boxes of course some insights of why a model is performing a certain way is lost, but as the study tries to cover as much data and algorithms as possible it would be unreasonable to go into the inner workings of all the algorithms. Thus the focus is rather on equally implementing and measure performance in relation to the other implemented models.

7.4 Ethical Aspects

When data about individuals is processed the variables should always be aggregated to the point of not being able to locate individual citizens. An “anonymous” data set, for instance, may easily cease to be anonymous if it includes variables that allow relatively unique individuals to be identified (Meyer, 2018). All the data that were used in this study were open source datasets, they did not contain any variables which could be combined to locate any individual. This i certain because the dataset were reviewed and no information about location, such as zip-codes or county codes were present in any of the dataset.

Another important point made by Meyer (2018) is that any researcher who publishes should be prepared to share the data used. This for the purpose of allowing other researches to reproduce results to validate the rigor of the research. Therefore, all data used are available as .csv files in the appendix.

“It is past time for the research community to realize that participants typically also expect that the data they contribute will be used to advance scientific truth, not merely to make scientific claims that cannot be verified.” (Meyer, 2018).

7.5 Future Work

Future work could build upon this structure of testing models against each other on different types of data, by incorporating more algorithms and a greater number of datasets. This would benefit the field by a broader view of how the different models perform on a broader range of datasets.

The study can also be extended by performing more extensive pre-processing of the datasets. This meaning a more flexible approach to preparing the datasets for the machine learning process, by evaluating feature importance or individual data points to minimize the amount of noise in the data. It is then of great importance that all pre-processing is documented, enabling reproduction of the results.

To improve upon the performance of the models, more extensive hyperparameter tuning can be performed. This would benefit the results by presenting the algorithms at their full potential, instead presenting them at equal effort of optimization.

7.6 Conclusion

Based on the results we can safely conclude that undersampling the data to improve classification performance is the most efficient method. This was true for all the classification algorithms for all of the datasets. However, undersampling does remove some of the original data, thus removing some of the data's power the reflect the real-world observation. Another conclusion proven by the results is the fact that data manipulation *(both undersampling and* oversampling*)* always improves algorithm performance. This was always true, over all algorithms and datasets.

As for model performance it is important to note that all the results are under the predefined limitations on the hyperparameter tuning and optimization, where randomized search was the only method for tuning the hyperparameters for all algorithms. So all conclusions stated around the algorithms are also bound to the setting in which these were implemented.

Haixiang et al. (2017) suggest the use of ensemble-based algorithms to account for biases caused by imbalanced datasets, this statement was proven true in case of this study with the datasets used. Both implemented ensemble classifiers *(GBDT and RF)* performed well with imbalanced data. We can therefore conclude that both ensemble learners were very versatile and could handle different shapes and size of the datasets.

The results also point towards MLP being the best performing algorithm when the dataset contains a lot of categorical columns, where the other algorithms performed worse. Furthermore, the algorithm showed keenness to overfit when presented with highly imbalanced data or big datasets in general, but MLP together with undersampling was a well performing combination on the dataset containing categorical features.

They also claim that applying a single specific classifier to tackle different kinds of imbalanced-data was an inefficient approach, because such a classifier would need to be tuned accordingly to the data iteratively thus making a *one for all solution like that very inefficient* (Li et al., 2016).

References

Amani, F. A., & Fadlalla, A. M. (2017). Data mining applications in accounting: A review of the literature and organizing framework. *International Journal of Accounting Information Systems*, *24*, 32-58.

Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. ISPRS Journal of Photogrammetry and Remote Sensing, 114, 24-31.

Biau, G., & Scornet, E. (2016). A random forest guided tour. Test, 25(2), 197-227.

Bose, I., & Mahapatra, R. K. (2001). Business data mining—a machine learning perspective. *Information & management*, *39*(3), 211-225.

Drummond, C., & Holte, R. C. (2003). C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In ICMLW.

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* (pp. 161-168).
Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). SMOTEBoost: Improving prediction of the minority class in boosting. In European conference on principles of data mining and knowledge discovery (pp. 107-119). Springer, Berlin, Heidelberg.

Claesen, M., & De Moor, B. (2015). Hyperparameter search in machine learning. arXiv preprint arXiv:1502.02127.

Datta, S., & Arputharaj, A. (2018). An Analysis of Several Machine Learning Algorithms for Imbalanced Classes. In 2018 5th International Conference on Soft Computing & Machine Intelligence (ISCMI) (pp. 22-27). IEEE.

Eklund, T., Back, B., Vanharanta, H., & Visa, A. (2002). Assessing the feasibility of self organizing maps for data mining financial information.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern recognition letters, 27(8), 861-874.

Foody, G. M. (2004). Supervised image classification by MLP and RBF neural networks with and without an exhaustively defined set of classes. International Journal of Remote Sensing, 25(15), 3091-3104.

Ghahramani, Z. (2003). Unsupervised learning. In Summer School on Machine Learning (pp. 72-112). Springer, Berlin, Heidelberg.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications, 73, 220-239.

Hand, D. J. (2007). Principles of data mining. *Drug safety*, *30*(7), 621-622.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence) (pp. 1322-1328). IEEE.

Huang, C., Li, Y., Change Loy, C., & Tang, X. (2016). Learning deep representation for imbalanced classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5375-5384).

King, G., & Zeng, L. (2001). Logistic regression in rare events data. Political analysis, 9(2), 137-163.

Kou, G., Lu, Y., Peng, Y., & Shi, Y. (2012). Evaluation of classification algorithms using MCDM and rank correlation. International Journal of Information Technology & Decision Making, 11(01), 197-225.

Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In Artificial Intelligence in Design'96 (pp. 151-170). Springer, Dordrecht.

Li, H., & Wong, M. L. (2015). Financial fraud detection by using Grammar-based multi-objective genetic programming with ensemble learning. In 2015 IEEE Congress on Evolutionary Computation (CEC) (pp. 1113-1120). IEEE.

Ling, C. X., & Sheng, V. S. (2008). Cost-sensitive learning and the class imbalance problem. Encyclopedia of machine learning, 2011, 231-235.

Maalouf, M., & Trafalis, T. B. (2011). Robust weighted kernel logistic regression in imbalanced and rare events data. Computational Statistics & Data Analysis, 55(1), 168-183.

McCue, C. (2014). *Data mining and predictive analysis: Intelligence gathering and crime analysis*. Butterworth-Heinemann.

Meyer, M. N. (2018). Practical tips for ethical data sharing. Advances in Methods and Practices in Psychological Science, 1(1), 131-144.

Naman D. Singh, Abhinav Dhall. (2018). Clustering and Learning from Imbalanced Data, ArXiv.

Provost, F. (2000). Machine learning from imbalanced data sets 101. In Proceedings of the AAAI'2000 workshop on imbalanced data sets (Vol. 68, pp. 1-3). AAAI Press.

Ravanshad, A. (2018). Gradient Boosting vs Random Forest. The Medium. Retrieved from: https://medium.com/@aravanshad/gradient-boosting-versus-random-forest-cfa3fa8f0d80.

Saberian, M., Delgado, P., & Raimond, Y. (2019). Gradient Boosted Decision Tree Neural Network. arXiv preprint arXiv:1910.09340.

Si, S., Zhang, H., Keerthi, S. S., Mahajan, D., Dhillon, I. S., & Hsieh, C. J. (2017). Gradient boosted decision trees for high dimensional sparse output. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 (pp. 3182-3190). JMLR. org.

Tan, B., Salakhutdinov, R., Mitchell, T., & Xing, E. (2019). Learning Data Manipulation for Augmentation and Weighting.

Tsai, C. F., Lin, W. C., Hu, Y. H., & Yao, G. T. (2019). Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. Information Sciences, 477, 47-54.

Worchester, P. (2019) A Comparison of Grid Search and Randomized Search Using Scikit Learn. The Medium. Retrieved from: https://blog.usejournal.com/a-comparison-of-grid-search-and-randomized-search-using-scikit-learn-29823179bc85.

Zanaty, E. A. (2012). Support vector machines (SVMs) versus multilayer perception (MLP) in data classification. Egyptian Informatics Journal, 13(3), 177-183.

Zhu, T., Lin, Y., Liu, Y., Zhang, W., & Zhang, J. (2019). Minority oversampling for imbalanced ordinal regression. Knowledge-Based Systems, 166, 140-155.

Zou, Q., Xie, S., Lin, Z., Wu, M., & Ju, Y. (2016). Finding the best classification threshold in imbalanced classification. Big Data Research, 5, 2-8.

Appendix

A – Creditcard.csv, https://drive.google.com/file/d/12a-hqIKahLPkSbEG_gW-SBYhKkj24Djp/view?usp=sharing

B – Companies.csv, https://drive.google.com/file/d/12iiV1KyrOgvzvuOnM5ZuHmpn9Xr6Ui6b/view?usp=sharing

C – Customers.csv, https://drive.google.com/file/d/12zUgoB70TpHO5-ij9oIjH9ODWONMEmlU/view?usp=sharing